Discussion on MTG Toolbox

MTG Toolbox

Quick Overview

- Reaching beyond the research community

- Making our work accessible and understandable to non-experts by lowering technical barriers

- With the aim to:

  - Involve those most impacted by and/or those most relevant to the research

  - Invite more diverse collaborations beyond the industry

(TL;DR) A platform for sharing tools with non-experts in an easily accessible manner

Plan for the meeting

Discuss and brainstorm

- All of us have different networks, and may have quite different perspectives on **what and how our research should be shared beyond academic circles**

- **What's missing** from the website in it's current state

- Share some **recommendations / resources**

What tools should we aim to share?

Sharing **tools** with non-expert users in an easily accessible manner

- Any tool aimed at creative applications of music technology

- Does the tool need to be generative and/or AI-based?

  - We shouldn't focus the platform on these applications.

  - Many of the topics we work on have the potential for broad creative applications — it's often just a matter of reimagining how they can be used in creative contexts.

Who should we design the tools for?

Sharing tools with **non-expert users** in an easily accessible manner

- Expertise here only refers to expertise in our own research areas

- We can (and sometimes should) assume that users will be experts in their own fields

- So, we shouldn't aim to develop tools that suit everyone!

- But, we should develop the tools in a way that :

  - Accessing them doesn't require technical knowledge of our research process and involved technologies

  - Ideally the tool is designed according to the habits/requirements of the target user group

How should we improve accessibility?

Sharing tools with non-expert users in an **easily accessible manner**

- Perhaps this is quite dependant on the target users and the target application

- That said, maybe regardless of the tool, we should adhere to some common practices

  - Deployment in a relevant and accessible manner

  - Proper documentation

    - Consider user's domain knowledge, and prepare documentation accordingly

    - (Ideally) Avoid engineering explanations. That said, try to convey the research method in non-technical terms

  - Demos

    - Show basic use cases

    - If possible, propose (to encourage) alternative ways of using the tool

  - Support

    - Provide proper channels for receiving feedback and providing support

Accessible Deployment

- **Standalone Apps**

- **VST plugins**

  - HARP (https://par.nsf.gov/servlets/purl/10543349)

  - NeuralMidiFx (https://neuralmidifx.github.io/)

- **Puredata**:

  - Plugdata for easy DAW integration (also builtin tools for exporting certain patches into VST plugins)

  - nn~ for deployment of torch models

- **Max**:

  - M4L for easy integration in Ableton

  - RNBO for creating VST plugins or Max Externals from max patches (https://rnbo.cycling74.com/)

  - nn~ for deployment of torch models

- **VCV Rack Modules**

- **Add-ons** for Audacity, Sibelius, Reaper, …

  - ReaScript for custom Reaper Scripts

  - DL Tools for Audacity (https://interactiveaudiolab.github.io/project/audacity.html)

How to decide on the deployment environment?

- Consider:

    - Target users' workflows

    - What you can afford

- Find a balance and deploy accordingly

- Ideally (if possible and affordable), multi-platform without requiring specialized hardware.

- In any case, do your best make the installation as easy, streamlined and familiar as possible

Who maintains the shared tools?

- The main developers of the tool

    - in whatever shape or form they prefer, and

    - of course to the extent they can afford to do so

- If the project is open-source, hopefully external collaborators would be encouraged to help with maintenance as well

- I can also help with development (deployment), content preparation and maintenance of projects

Sharing Projects Post-Deployment

Content for Website

What to share?

- **Overview**

- **Documentation**

- **Download**

- **Examples (Demos)**

- Bugs & Features

- Learn More

- **License**

- **Contact Developers**
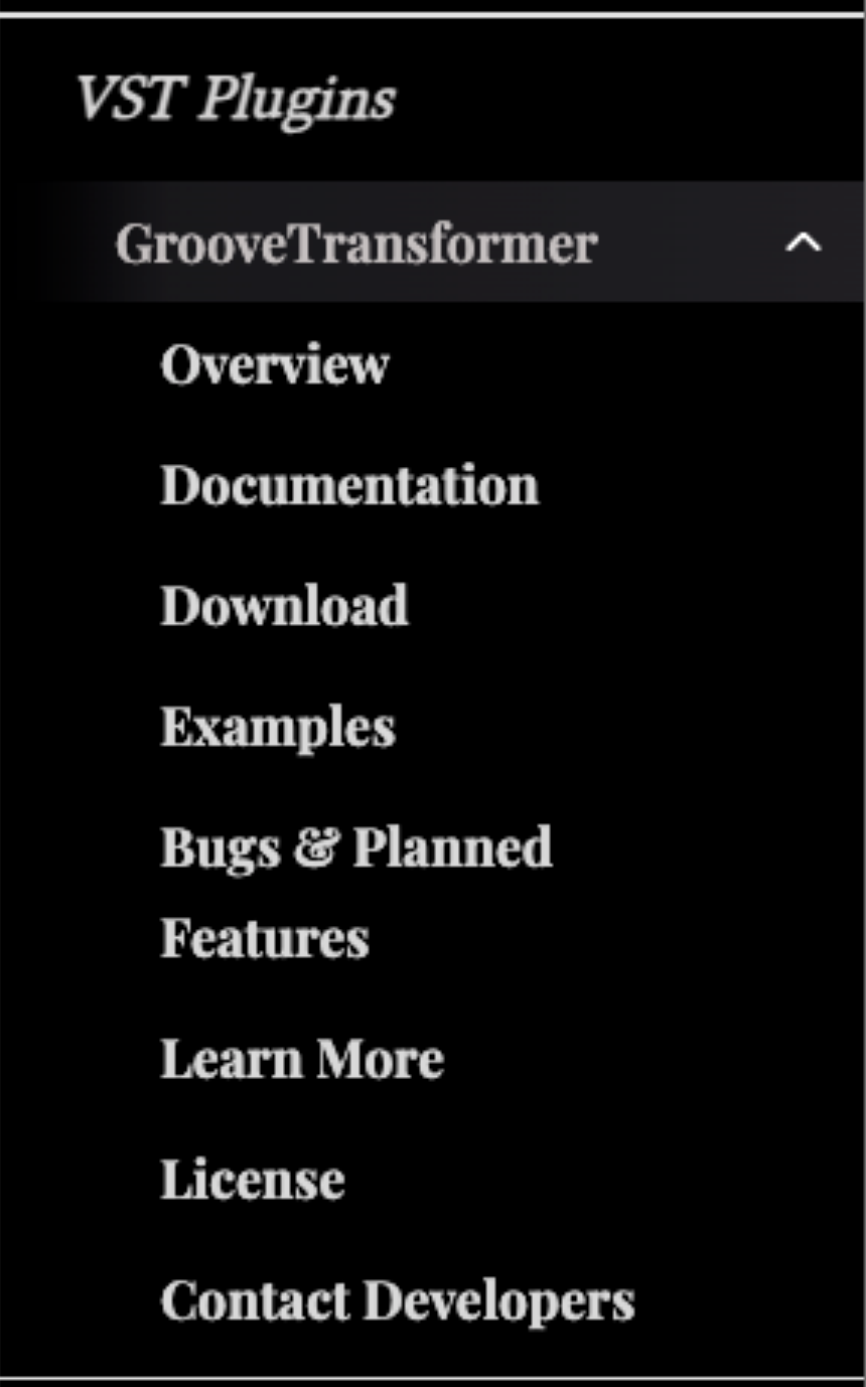
Bold: Possible Must-haves (?)

Content for Website
Overview

- Explain the **concept** behind the tool

- Explain most important aspects / usages of the tool

- Explain the **technical aspects** of the tool **in non-technical terms**

- Use text / visuals / videos, whichever you prefer and are more comfortable with
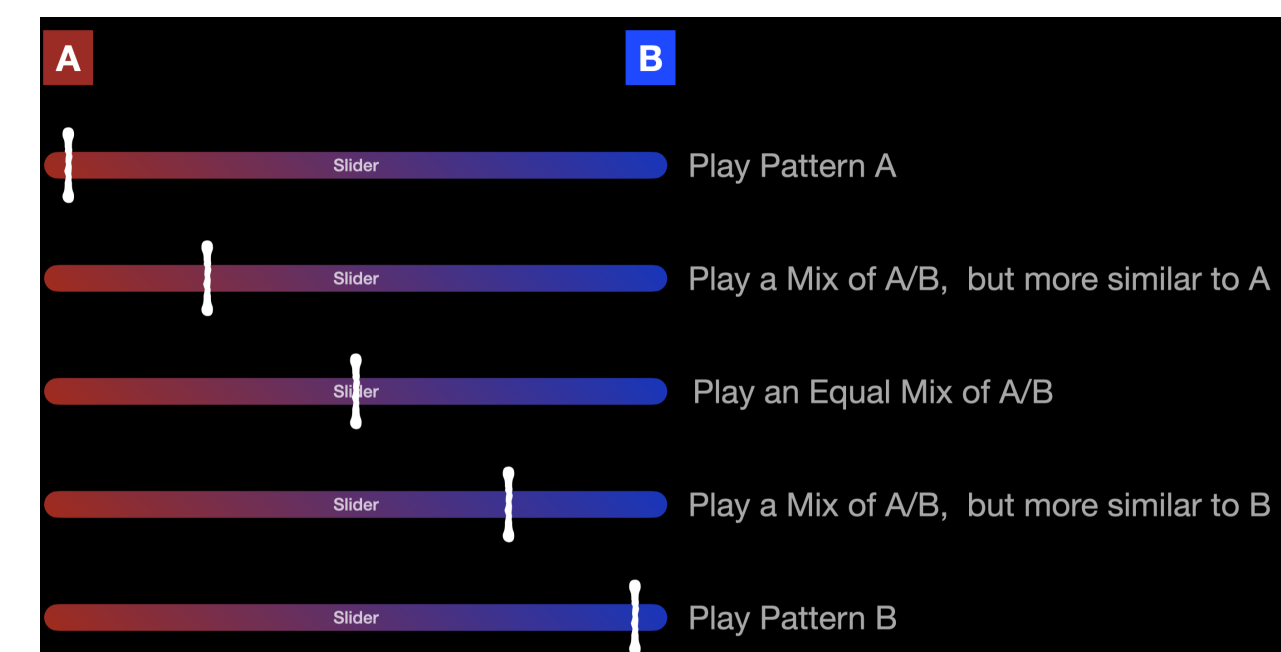
Content for Website

Documentation

- Installation / Setup Guide

- Usage Manual

- Be as detailed and thorough as possible

- Use terminologies familiar to the target group

Documentation

A detailed documentation of GrooveTransformer VST

Content for Website
Download

- Provide Links to download the tool

- Specify if specific requirements is needed (platform, ...)

- You can host the tool anywhere, and just provide links here

- If you can, provide access to source code!

Download

VST Installation Files for GrooveTransformer

TABLE OF CONTENTS

Version 0.0.1

See the following tables for the download links for the GrooveTransformer VST plugin.

-	Download Links
Windows	**Download**
MacOS (ARM)	**Download**
MacOS (Intel)	**Download**
Linux	**Download**

Plugin Source Code

The plugin source code can be found **here**.

The plugin was developed using **NeuralMidiFx**, a wrapper we developed for streamlining the deployment of generative models within VST plugins.

Content for Website

Download

- Where to host?

  - Github Releases

  - https://patchstorage.com/

  - https://maxforlive.com/library/

  - https://gumroad.com/

Content for Website
Examples / Demos

- Prepare audio / video demos

- Cover the basic usages

- If possible, provide more advanced (alternative) uses that users may not be aware of

**Examples**

Where and how to use the GrooveTransformer VST

🔗 TABLE OF CONTENTS

Content for Website
Bugs & Planned Features

- Track identified issues to be fixed

- Specify if a new feature is to be implemented in future

- Do we need this?

  - Does it improve transparency?

  - Does it help with emphasizing that projects
    are maintained by small teams
    hence user's shouldn't expect tools to be
    on par with commercial products?

- Use this page for tracking bugs and feature plans. For reporting bugs,
  refer users to "Contact Developers" page
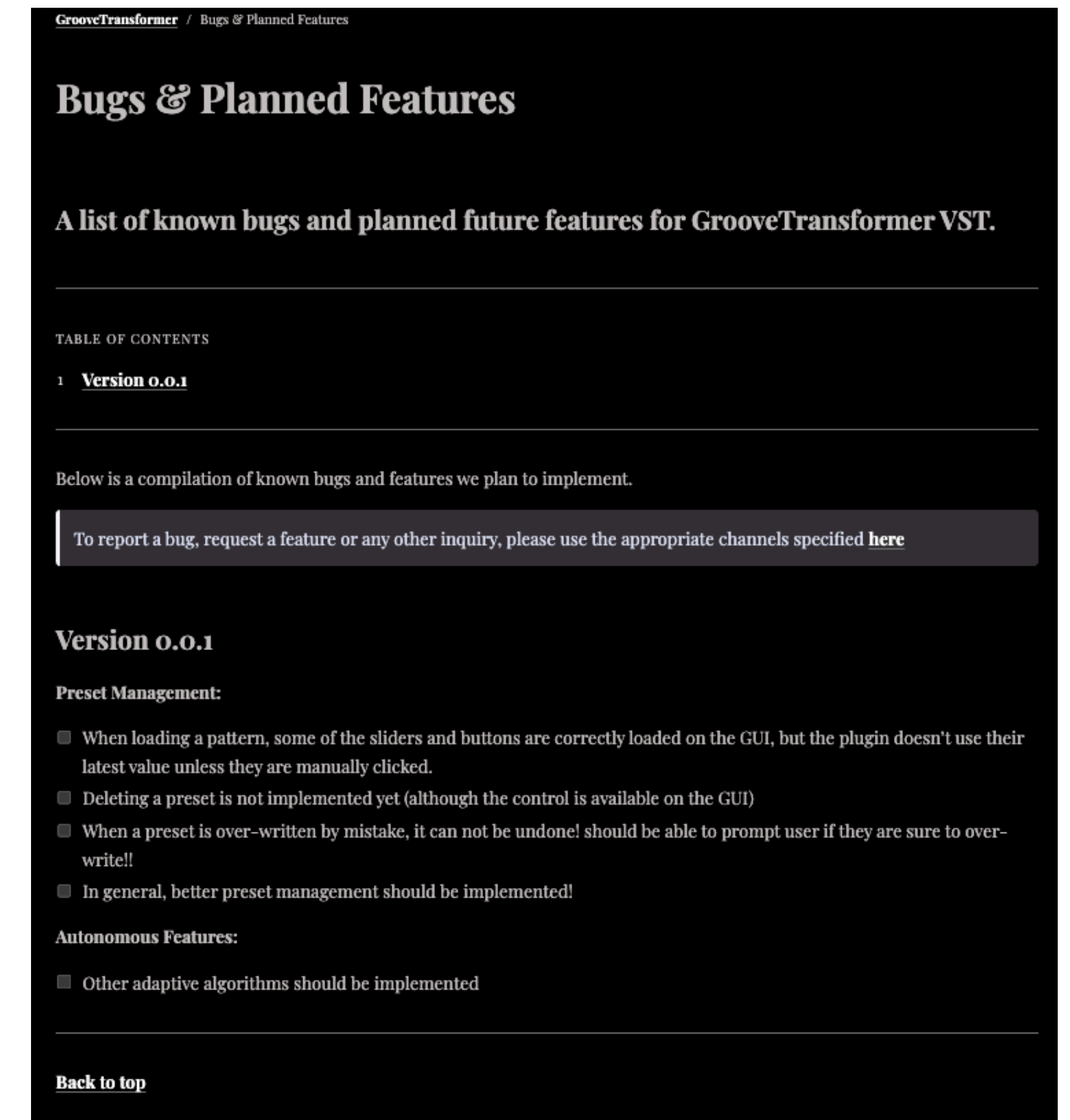
Content for Website
Learn More

- Provide additional info

  - Technical documents (papers / source code / ...)

  - Showcases

  - Any other resource you want to share / promote

Content for Website

License

Content for Website

Contact Developers

- Mention the people behind the project

- Acknowledge Fundings / External Collaborators

- Provide channels for communication

Contact Developers

Contact the developers for support, feedback, or feature requests.

TABLE OF CONTENTS

1 Developers
2 Acknowledgements
3 Support, Bug Reports, and Feature Requests
4 Feedback, and Direct Contact
5 Collaborations

Content for Website

Contact Developers

- You decide how you want to receive communications?!

    - Is Github issues a good place?

    - If you're ok with direct contact, provide email

- Do we need a specific forum perhaps?

Support, Bug Reports, and Feature Requests

Please use the **GitHub Issues** page.

Feedback, and Direct Contact

If you prefer to contact us directly, please contact **Behzad Haki**.

Collaborations

If you are interested in collaborating with us, please contact any of the developers above.

Content for Website

What should be mandatory?

- **Overview**

- **Documentation**

- **Download**

- **Examples (Demos)**

- Bugs & Features

- Learn More

- **License**

- **Contact Developers**

How should we organize the tools on the platform?

- By task? (e.g. synthesis, FX, …)

- By Modality (e.g. Audio/Midi/Text-to-, …)

- By Deployment Type (e.g. VSTs, Max, Puredata, Webpage, …) —> This is the current approach

- Or should we simply organize by the order in which tools were added to the platform?

Do you have any ideas for future tools?

Do you want to be involved in this project?